**CSC 5750 – Principles of Web Technology**

**Homework 3 – [your name]**

**25 points – Due March 1, 5pm**

**Late deadline is March 3, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Gather the following files into a ZIP file:

● This document

● Inventory Manager page (.HTML)

● Inventory Summary page (.HTML)

● CSS file (.CSS)

● Any other optional files specific to your implementation

**d)** Submit the ZIP file to the Blackboard item where you downloaded this document.

You're continuing to operate your storefront business to sell products and/or services. Now you need to add a web page to provide an inventory summary for the business. Enhance the web site you created in Homework 2 with the following changes. You may start with your Homework 2 or the Homework 2 key on Blackboard.

**Inventory Manager page**

● Add ", v3" to the end of the page title and heading. This will indicate that this is version 3 of the web site.

● Change the name of function loadComboBox to loadProductData. Add logic to check for the existence of the products array in local storage. When a getItem is performed on a local storage item that doesn't exist, null is returned. If the products array is not in local storage, do nothing since the page will use the already-declared products array. If the products array is in local storage, parse it and overlay the already-declared products array. The purpose of this change is to insure that you're using the latest version of your products data.

● Add logic such that changes to inventory (orders and sales) are saved in the products array. Also, save the products array in local storage.

● Add the following control:

● **Summary** button – this button triggers a function that opens the Inventory Summary page.

**Inventory Summary page**

Add an **Inventory Summary page** to enable your staff to view an inventory summary.

● Create **function loadProductData** that retrieves the products array from local storage. It then loops through the array and counts the total inventory of all products and the total value of the inventory. Finally, it shows values on the page. When the page loads, the function will be called:

<body onload=" loadProductData()">

● Add the following controls to the page:

● **Product count:** label and read-only text box – this shows the number of distinct products in inventory.

● **Product inventory count:** label and read-only text box – this shows the total inventory count of all products.

● **Value ($):** label and read-only text box – this shows the total inventory value of all products.

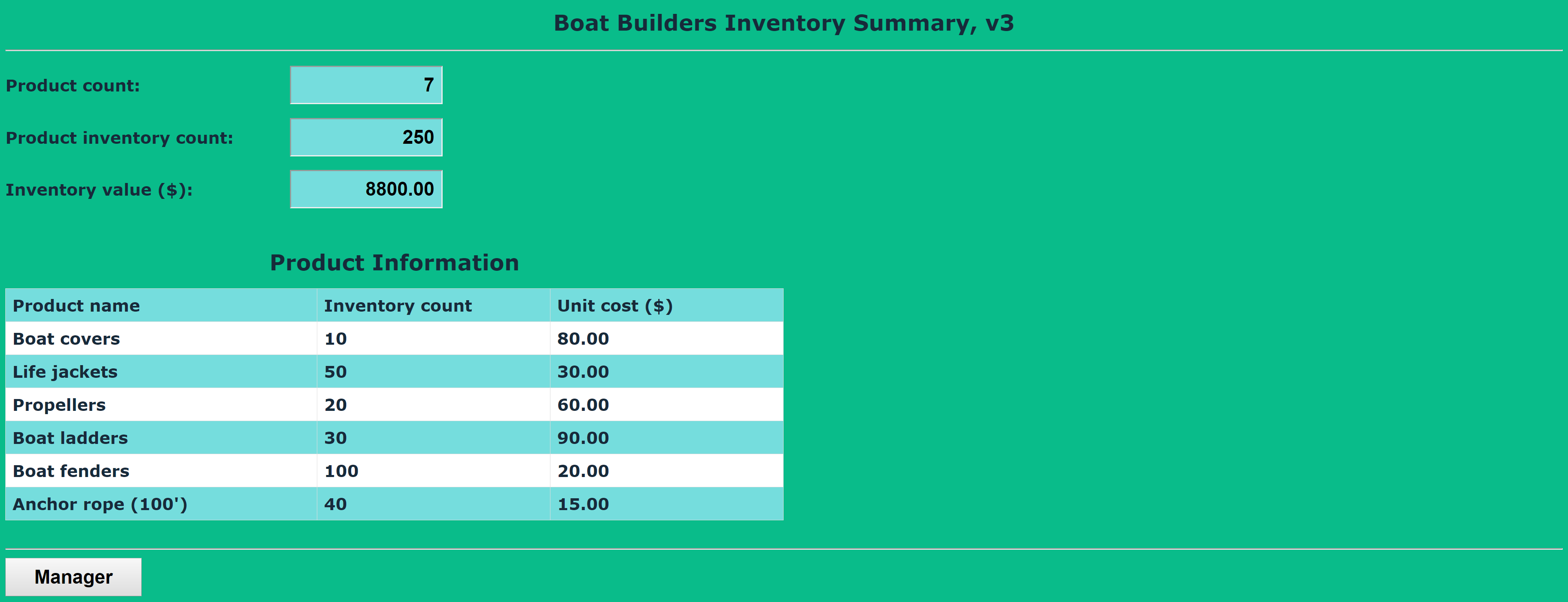
● **Manager** button – this button triggers a function that returns to the Inventory Manager page.

Here is what the page might look like.



**Extra credit [5pts]**

Add a table to the page that shows product information. This must be created using document object model (DOM) commands. The commands may be called as part of function loadProductData. Here is what the page might look like.



**Inventory Manager page**

*[your Inventory Manager page HTML code here]\**

**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”**

**Bold**

*[your Inventory Manager page screenshots here]\*\**

**Inventory Summary page**

*[your Inventory Summary page HTML code here]\**

**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”**

**Bold**

*[your Inventory Summary page screenshots here]\*\**

\* **Copying-and-pasting web page code to a Word document**

1) From the HTML editor window, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting application output to a Word document**

1) From the web page, maximize the browser window.

2) From the browser window, press **ALT-PrintScreen**.

3) From within the Word document, press **CTRL-V**.